

UNIVERSITY OF BAHRAIN  
COLLEGE OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SCIENCE

ITCS490 – Software Engineering II

Midterm Exam

Semester I

Wednesday 13-4-2016 @ 14:30

Time Allowed: 1.30 Hours

Student I.D.
Student Name
Section

Question 1	15	14
Question 2	25	24
Question 3	25	22.5
TOTAL	65	60.5

Notes:

- Make sure you have 7 pages including the cover Page
- If you use any extra space (on the back of the pages) mark the question numbers clearly.
- Make sure you write your Name, ID# and Section number clearly.
- If you have a mobile phone please switch it off before the test starts.
- It is yours responsibility to hand over the answer sheet to the invigilation staff.

1. Compare between the RAD and incremental process models using the following:

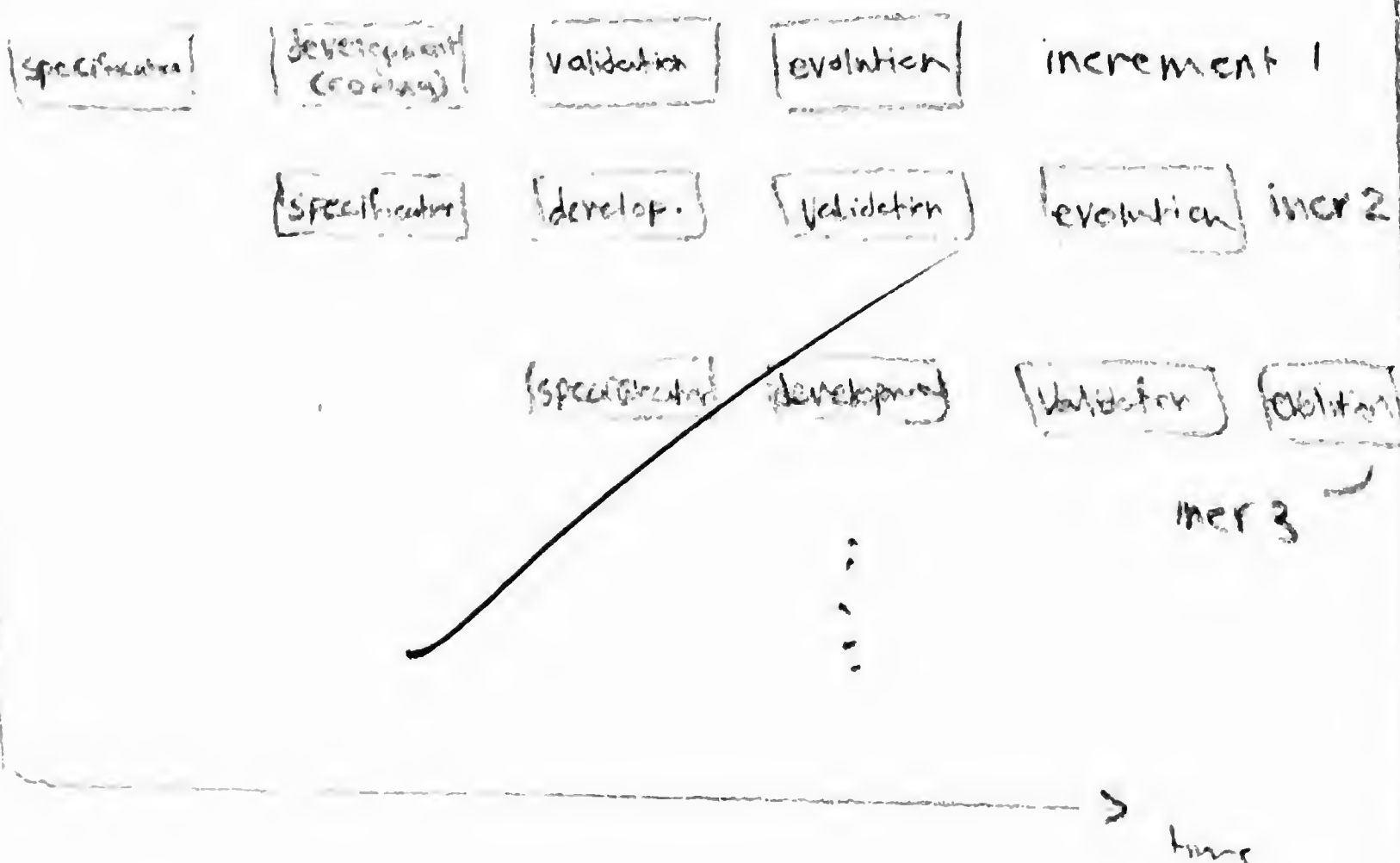
- Draw both models,
- Complexity,
- Technical risk
- Size of the project,
- Required time to finish, and
- Size of the team members.

14  
15

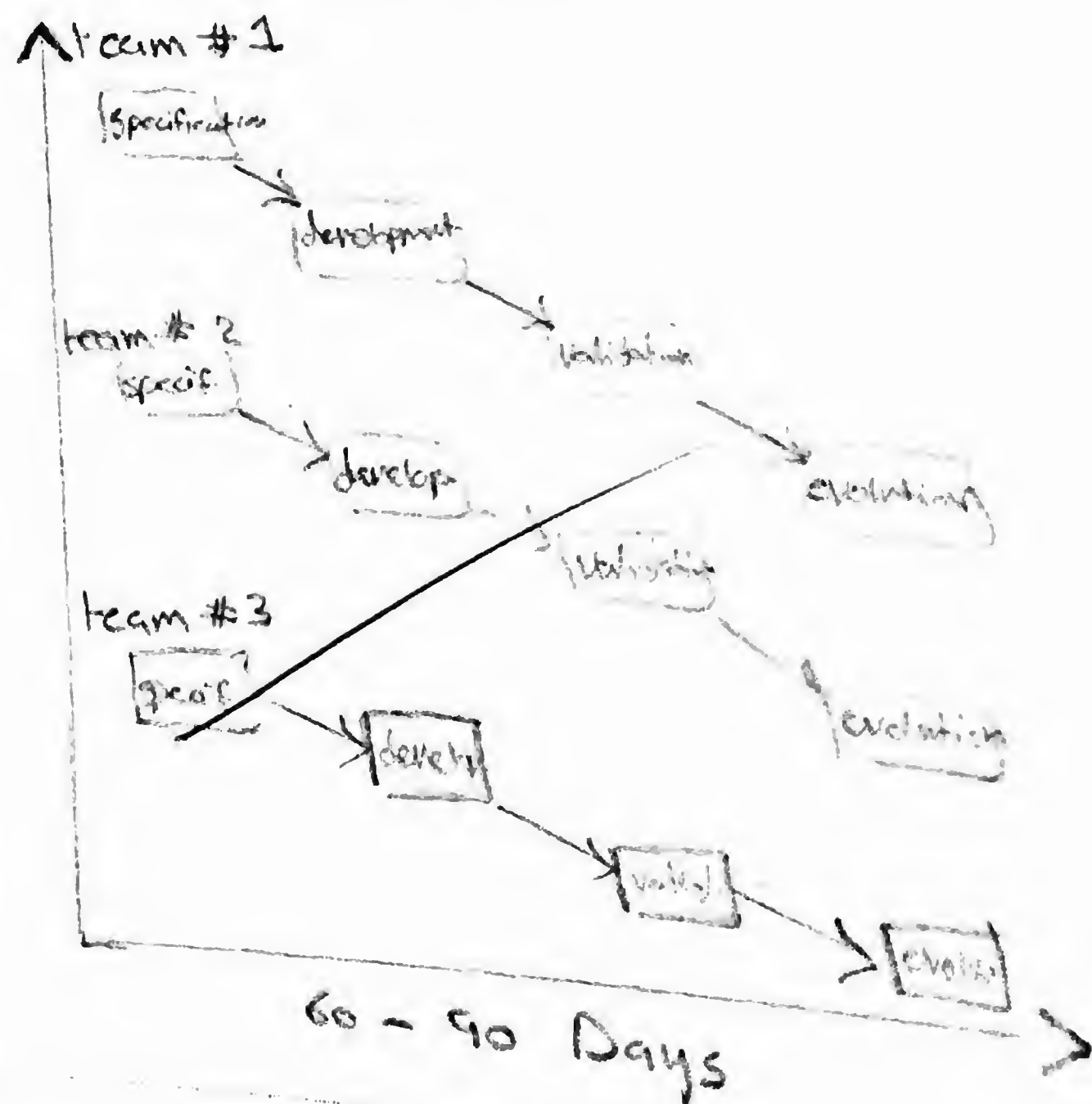
Comparison :

- Complexity: incremental is for more complex systems, where the system is delivered as increments. RAD is for less complex systems than in incremental.
- Technical risk: RAD can't be used when technical risks are high. ~~Incremental is better in terms of considering technical risk.~~
- Size of the Project: incremental can take larger projects than RAD. both of them can take large projects.
- Required time to finish: RAD: 60-90 Days. Incremental: open time.
- Size of the team members: RAD: require large number of team members to finish fast, all working at the same time. Incremental: used when part of the team (staff) can't be available all time (less size than RAD) one team only

### Incremental Process model



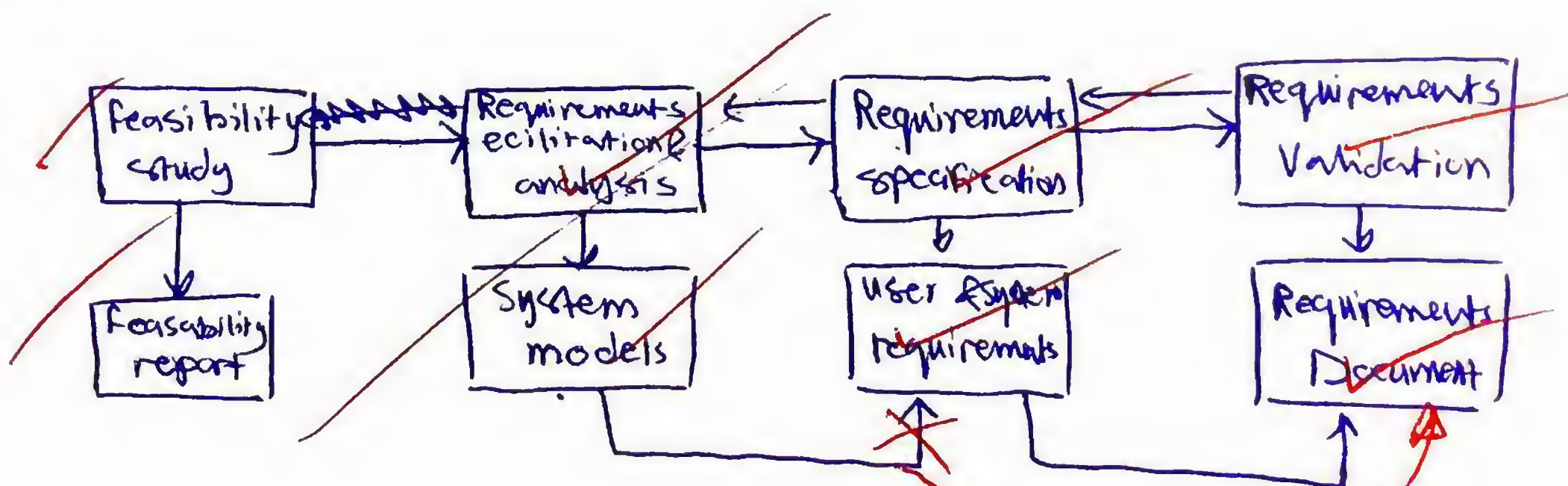
### RAD Process model





## 2. Answer the following questions [25 marks]

1. If a system for a library is required, analyst must find out how the library is operated. Is it:
  - a. Domain understanding
  - b. Requirement classification
  - c. Conflict resolution
  - d. None of the above
2. Traceability is concerned with the relationships between requirements, their sources and system design.
  1. True
  2. False
3. Non-functional requirement may lead to a functional requirement.
  1. True
  2. False
4. Domain requirements are the requirements that come from the application domain of the system and that reflect characteristics of that domain. They are only non-functional requirements.
  1. True
  2. False
5. One of the non-functional requirements was written as "Due to government policy, the information stored in the database should not be disclosed to a person other than to whom it belongs to". What category this requirement belongs to?
  - a. Product requirements
  - b. Organizational requirements
  - c. External requirements
  - d. None of the above
6. Which of the following techniques is more suitable/strong for requirements elicitation/requirements gathering for a new system?
  - e. Brainstorming sessions
  - f. Interviews
  - g. Task analysis
  - h. Form analysis
7. Draw the requirement engineering process and explain why feasibility assessment is out side the iteration process.



Feasibility study is to study if the system is worthwhile: does it achieve the organization goal, is it with the budget and current technology & if it can integrate with other systems. So if not, it will not be built, so no need for iterative.



You are working on insurance management system; write 4 core functional requirements and non-functional requirements.

### [A] Functional Requirements

1. The system will allow the user (client, staff, manager) to login to the system.
2. The system will allow the client to get a quotation.
3. The system will allow the staff to approve client documents.
4. The system will allow the manager to generate different types of reports.

### [B] non-functional Requirements

1. Security: the system must be secure in the login part; by user name and password that has constraints. it must provide different services to different types of users (client, staff, manager) (Privileges).
2. Performance: the system must generate different types of reports in few seconds. (100 record per second).
3. Usability: the system must provide user-friendly interface, easy to use & easy to remember.
9. Traceability is concerned with the relationships between requirements, their sources and the system design. What are the types of traceability of information that should be maintained?

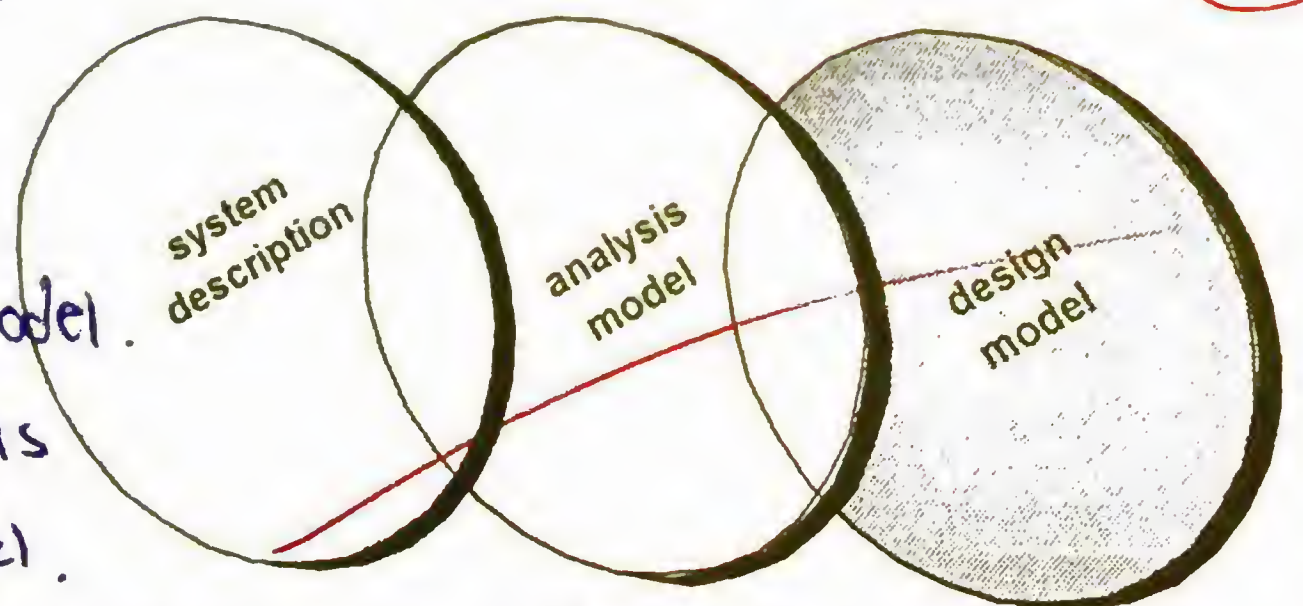
- 1) requirements traceability
- 2) sources traceability
- 3) design traceability

3

10. Why we consider the analysis model is the bridge between system description and design model?

The analysis models! produce the different system models (Data Flow Diagram, ERD, State Transition Diagram), which are used as sources to do the design in design model. for example,

- interface design in design model uses DFD as a source for building from the analysis model.
- The source of designing database is the ER diagram and data declaration in the analysis model.



4



Answer the following questions [25 marks]

1. What are the characteristics of a good design?

- 1) it should be readable and understandable. ✓
- 2) Consider designing : data design, architecture design, interface design, procedural design. ✓
- 3) Apply functional independence in activities and loosely coupled. ✓
- 4) built in iterative way. ✓
- 5) Apply design concepts :
  - abstraction
  - modularity
  - refinement
  - architecture
  - control (call & return, partitioning)
  - Information hiding : information in one module is inaccessible by other module that does not need it.
- 6) consider different types of requirements. *explicit & implicit requests.*

Consider the following code segment of two modules.

```
float salary(Employee y) {
    :
    code to calculate and return the total salary of employee y
    :
}
```

```
void PrintSalarySlip(Employee y) {
    :
    float NetSalary = salary(y);
    cout << "Net Salary = " << NetSalary << endl;
    :
}
```

Coupling :

- data
- stamp
- control
- ~~content~~
- common
- Content

2. What type of coupling between above two models and why

2 ✓ Stamp coupling, because <sup>in</sup> PrintSalarySlip (salary(y)) calls the other function and sends data structure y of type Employee.

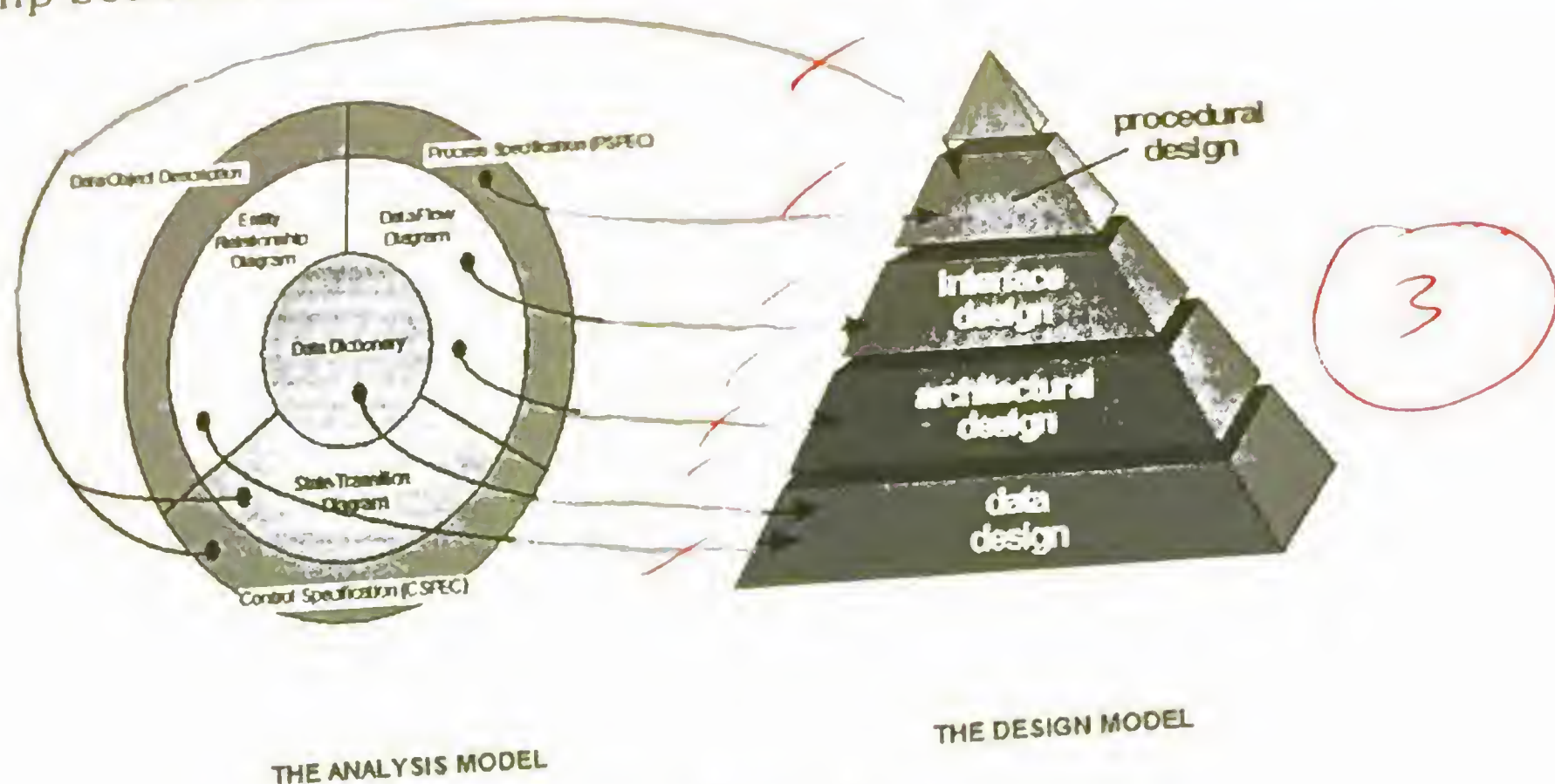
3. How we reduce the coupling between modules by:

- a. eliminate unnecessary relationships. ✓
- b. reduce the number of necessary relationships.
- c. loosing the tightness of necessary relationships. *easing*

2-5



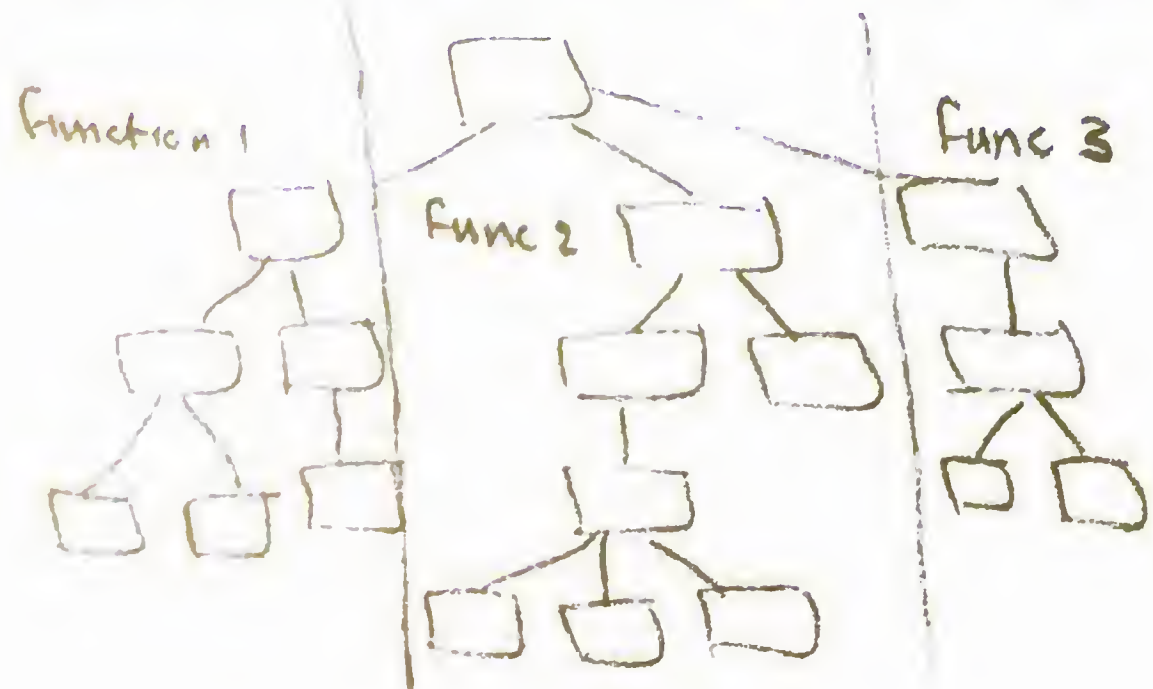
2. Find the relationship between the analysis model and the design model in the following figure.



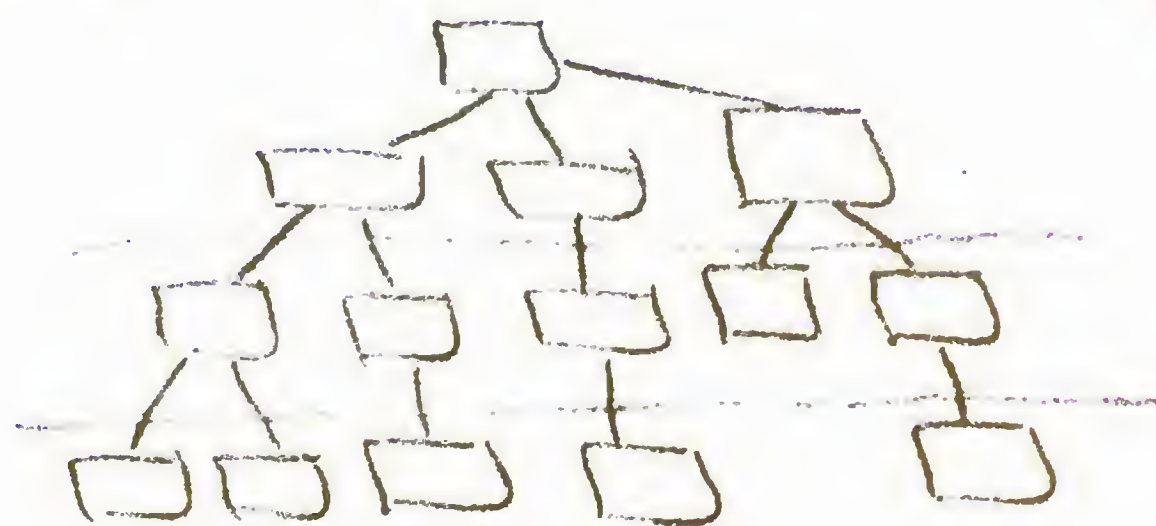
- The source of designing the database is the data dictionary DFD and ER diagram.  
 a. True  
 b. False ✓
- The source of designing database is the ER diagram and data dictionary in the analysis model.  
 a. True ✓  
 b. False
- The source of interface design in the analysis model is the DFD and ER diagram.  
 a. True  
 b. False ✓
- Information hiding leads to divide-and-conquer strategy.  
 a. True  
 b. False ✓
- Independence is measured using two qualitative criteria which are cohesion and coupling.  
 a. True ✓  
 b. False
- Differentiate between horizontal partitioning and vertical partitioning in Structural Partitioning.

horizontal Partitioning: for describing each function of system !

vertical Partitioning: for decision making !



- horizontal Partitioning -



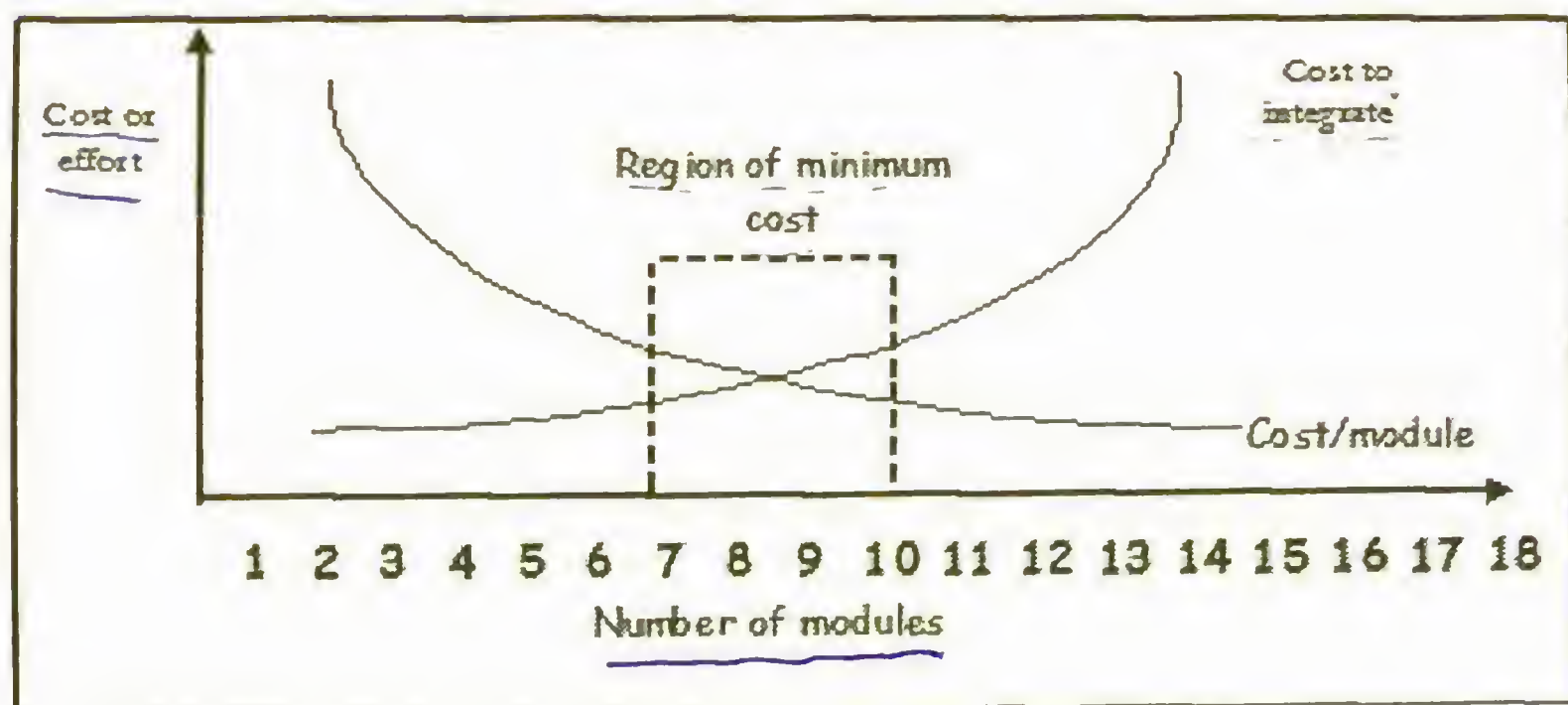
- vertical Partitioning -

please see your notes

2/4



3. Un-modularity and over-modularity are two terms that must be considered in designing systems. Look at the following figure and discuss the relation between integration and modularity from the point of cost of integration and the cost of building each module. [5 marks]



- in over-modularity the system is divided into large number of small sub-systems. Therefore it is easy to build ~~the~~ each of the small sub-systems (modules) with small cost for each. However, it will be very hard and costly to integrate the large number of small subsystem which shows clearly in the figure above.
- in Un-modularity the whole system is built as one module without dividing it. ~~like~~ The single big module is hard and costly to building it as one unit. However, there will be no cost for integration. (or in less modularity system, low cost for integration).
- So more modules  $\rightarrow$  more integration cost, less <sup>single</sup> module cost.  
less modules  $\rightarrow$  less integration cost, more module cost.